

### Numerical Integration Program (TI-82)

This program calculates left- and right-hand Riemann sums, and the trapezoidal, midpoint and Simpson approximations. Since there's not room on the calculator to label each approximation separately, we use a compressed method of displaying the results. For instance, the label LEFT/RIGHT indicates that the next two numbers are the left- and right-hand Riemann sums, respectively.

#### Notes:

1. Select 'PRGM' to get the program menu, move to 'NEW' to enter a new program. You must first give it a name (for example, 'INTEGRAL') when prompted; to finish entering/editing a program, hit 2nd QUIT.
2. The function to be integrated must be entered as  $Y_1$  (accessed by the "Y =" button). When  $Y_1$  occurs in a program, it is evaluated at the current value of  $X$ .
3. The lower limit of integration must be less than the upper limit.
4.  $IS > ($  means that the PRGM button must be pushed and then  $IS > ($  selected, not that  $I, S, >$  and  $($  are to be entered separately. 'Disp' and 'Input' are to be found under PRGM, I/O, pushed while entering a program.
5. To run a program, select PRGM, EXEC. To stop a program while it is running, hit ON.
6. Test the program by evaluating  $\int_1^3 x^3 dx = 20$ , using 100 subdivisions. You should get left- and right-hand sums of 19.7408 and 20.2608, respectively. For the trapezoid approximation, you should get 20.0008. For the mid point approximation, you should get 19.9996. For Simpson's rule, you should get exactly 20.

Name= INTEGRAL	<i>Where to Find The Commands</i>
:Disp "LOWER LIMIT"	Disp and Input are accessed via PRGM, I/O
:Input A	Enter lower limit of integration.
:Disp "UPPER LIMIT"	
:Input B	Enter upper limit of integration.
:Disp "DIVNS"	
:Input N	Enter number subdivisions.
:(B - A)/N → H	Stores size of one subdivision in H (Note that → means hit STO button).
:A → X	Start X off at beginning of interval.
:0 → L	Initialize L, which keeps track of left sums, to zero.
:0 → M	Initialize M, which keeps track of midpoint sums, to zero.
:1 → I	Initialize I, the counter for the loop.
:Lbl P	Label for top of loop. Lbl is accessed via PRGM, CTL.
:L + H * Y <sub>1</sub> → L	Increment L by Y <sub>1</sub> H, the area of one more rectangle. (Y <sub>1</sub> is accessed via Y-VARS, or 2nd VARS.)
:X + .5H → X	Move X to middle of interval.
:M + H * Y <sub>1</sub> → M	Evaluate Y <sub>1</sub> at the middle of interval and increment M by rectangle of this height.
:X + .5H → X	Move X to start of next interval.
:IS > (I, N)	IS > ( is accessed via PRGM, CTL. This is the most difficult step in the program: adds 1 to I and does the the next step if I ≤ N (i.e., if haven't gone through loop enough times); otherwise, skips next step. Thus, if I ≤ N, goes back to Lbl P and loops through again. If I > N, loop is finished and goes on to print out results. Continue here if I > N, in which case the value of X is now B.
:Goto P	Goto is accessed via PRGM, CTL. Jumps back to Lbl P if J ≤ N.
:Disp "LEFT/RIGHT" "L-R-T-M-S"	
:Disp L	L now equals the left sum, so display it.
:L + H * Y <sub>1</sub> → R	Add on area of right-most rectangle, store in R.
:A → X	Reset X to A.
:R - H * Y <sub>1</sub> → R	Subtract off area of left-most rectangle.
:Disp R	R now equals right sum, so display it.
:(L + R)/2 → T	Trap approximation is average of L and R.
<del>:Disp "TRAP/MID/SIMP"</del>	
:Disp T	Display trap approximation.
:Disp M	Display midpoint approximation.
:(2M + T)/3 → S	Simpson is weighted average of M and T.
:Disp S	Display Simpson's approximation.